

# ALMO

Objectifs et Structure du module  
Présentation du processeur MIPS

## Plan du cours

1. Pourquoi ce cours zéro
2. Contenu du module
3. Organisation pratique
4. Présentation du processeur MIPS



crédit : <http://coursz.com/definition-d-un-ordinateur>

## Pourquoi ce cours zéro

- ALMO signifie Architecture Matérielle et Logicielle des Ordinateurs  
C'est un sujet très vaste qui va de la programmation du processeur en assembleur (le langage de base) à la multi-programmation sur des ordinateurs multi-coeurs.
- Le module ALMO propose de répondre à des questions telles que :
  1. Que trouve-t-on dans un ordinateur ?
  2. Comment un programme fait pour lire des données depuis un clavier et envoyer ses résultats sur un écran ? ....  
... et en même temps ... :-)  
faire de la décompression d'image ou autre chose ?
  3. Comment exécuter rapidement un programme quand la mémoire est lente ?
- L'objectif de ce premier cours est de présenter les différents aspects abordés et de poser les questions auxquelles nous allons apporter des réponses au fil des semaines.

## Sujets abordés dans le module ALMO

### Dans l'ordre

- Architecture Matérielle et Logicielle d'un Ordinateur
- Programmation en assembleur du processeur MIPS
- Chaîne de compilation C
- Accès aux périphériques via les appels (au) système (d'exploitation)
- Organisation et gestion de la mémoire
- Hiérarchie des caches
- Structure d'un programme exécutable, chargement en mémoire
- Interaction du système d'exploitation avec les périphériques
- Multiprogrammation
- Virtualisation de la mémoire

## Architecture Matérielle et Logicielle d'un ordinateur

Un ordinateur contient :

- un ou plusieurs coeurs (unités de calcul)
- de la mémoire (et même une hiérarchie de mémoire)
- des périphériques (écran, clavier, disque, réseau, capteurs, etc.)
- une infrastructure de communication entre le(s) core(s), la mémoire et ses contrôleurs d'E/S

Un ordinateur exécute :

- une ou plusieurs *applications*
- sous le contrôle d'un système d'exploitation

## Architecture Matérielle et Logicielle d'un ordinateur

Un ordinateur contient :

- un ou plusieurs coeurs
  - de la mémoire
  - des périphériques
  - une infrastructure de communication

Un ordinateur exécute :

- une ou plusieurs *applications*
- sous le contrôle d'un système d'exploitation

Coeur (core)

composant exécutant les instructions du programme (séquence d'instructions)

- Est-ce que processeur et coeur sont synonymes ?
- Quel est le schéma de principe d'un coeur de calcul ?
- Quelles sont les types d'instructions ?
- Que veut dire RISC ou CISC ?
- Les instructions travaillent sur des registres, qu'est-ce qu'un registre ?
- En combien de temps s'exécute une instruction ?
- Comment fonctionne un ordinateur multi-coeurs ?
- Qu'est-ce qui limite le nombre de coeurs dans un ordinateur ?

## Architecture Matérielle et Logicielle d'un ordinateur

Un ordinateur contient :

- un ou plusieurs coeurs
- **de la mémoire**
- des périphériques
- une infrastructure de communication

Un ordinateur exécute :

- une ou plusieurs *applications*
- sous le contrôle d'un système d'exploitation

Mémoire

où sont enregistrés les instructions, les données et les contextes d'exécution

- Qu'est-ce qu'un bit, un octet, un mot, une ligne, une page ?
- Qu'est-ce qu'un espace d'adressage ?
- Qu'est-ce que la hiérarchie de la mémoire ?
- Que signifie cache de mémoire ?

## Architecture Matérielle et Logicielle d'un ordinateur

Un ordinateur contient :

- un ou plusieurs coeurs
- de la mémoire
- **des périphériques**
- une infrastructure de communication

Un ordinateur exécute :

- une ou plusieurs *applications*
- sous le contrôle d'un système d'exploitation

Périphériques

composants connectés au(x) coeur(s) réalisant des fonctions spécifiques

- Quels sont ces périphériques ?
- Comment sont-ils commandés ?
- Quelle différence entre un périphérique cible et un périphérique initiateur ?
- Comment un périphérique demande-t-il un service au système ?
- Que signifie ISR ?

## Architecture Matérielle et Logicielle d'un ordinateur

Un ordinateur contient :

- un ou plusieurs coeurs
- de la mémoire
- des périphériques
- une infrastructure de communication

Un ordinateur exécute :

- une ou plusieurs *applications*
- sous le contrôle d'un système d'exploitation

Infrastructure de communication :

utilisé par un coeur pour accéder à la mémoire et aux périphériques.

- Quels sont les types d'infrastructure de communication ?
- Qu'est-ce qu'un bus de communication ?
- Combien de coeurs peuvent utiliser la mémoire sans se gêner ?
- Quelle est la durée d'attente avant d'accéder à la mémoire ?

## Architecture Matérielle et Logicielle d'un ordinateur

Un ordinateur contient :

- un ou plusieurs coeurs (unité de calcul)
- de la mémoire
- des périphériques
- une infrastructure de communication entre le(s) core(s), sa mémoire et ses contrôleurs d'entrées-sortie.

Un ordinateur exécute :

- une ou plusieurs *applications*
- sous le contrôle d'un système d'exploitation

Applications

par exemple : un lecteur multimédia, un jeu, un traitement de texte.

- Qu'est-ce qu'un thread (fil d'exécution) ?
- Qu'est-ce qu'un processus ?
- Qu'est-ce qu'un contexte d'exécution ?

## Architecture Matérielle et Logicielle d'un ordinateur

Un ordinateur contient :

- un ou plusieurs coeurs (unité de calcul)
- de la mémoire
- des périphériques
- une infrastructure de communication

Un ordinateur exécute :

- une ou plusieurs applications
- sous le contrôle d'un système d'exploitation

Système d'exploitation

programme permettant l'exécution des applications et gérant l'allocation du matériel (coeur de calcul et mémoire) et l'accès aux entrées-sorties.

- Quels sont les services proposés par un système d'exploitation ?
- Comment garantir les privilèges du système d'exploitation ?
- Comment gérer l'exécution de plusieurs threads ?
- Qu'est-ce qu'un pilote de périphériques ?
- Comment une application peut-elle attendre un événement externe ?

## Ressources

- web : <https://www-soc.lip6.fr/trac/sesi-almo>
  - Programme du module et Enoncés de TD et TP
  - Configuration de l'environnement des TP
  - Documents
    - Architecture externe et Langage d'assemblage du MIPS
    - Code du système GIET
    - Enoncés des TD et TP (non autorisé à l'examen)
    - annales corrigées d'examens (non autorisé à l'examen)
- fichiers : /Infos/lmd/2018/licence/ue/3I004-2018oct/
  - code source des TP
  - code source du "système d'exploitation" GIET
  - binaires des outils
  - script d'environnement

## Outils

- Simulateur du processeur MIPS
  - mars
- Cross-compileur MIPS pour MARS
  - mipsel-mars-elf-gcc
- Simulateur de la machine
  - simul\_almo\_generic
- Chaîne de compilation pour la machine
  - mipsel-unknown-elf-as
  - mipsel-unknown-elf-gcc
  - mipsel-unknown-elf-ld
  - mipsel-unknown-elf-objump

## Evaluation

Ne pas venir en cours réduit fortement vos chances de succès  
Les TD doivent être préparés chez-vous !

- 1 TP noté par semaine (10 en tout) ⇒ 20% de la note
- 4 interrogations écrites pendant les TD ⇒ 20% de la note
- 1 examen composé de 4 parties ⇒ 60% de la note

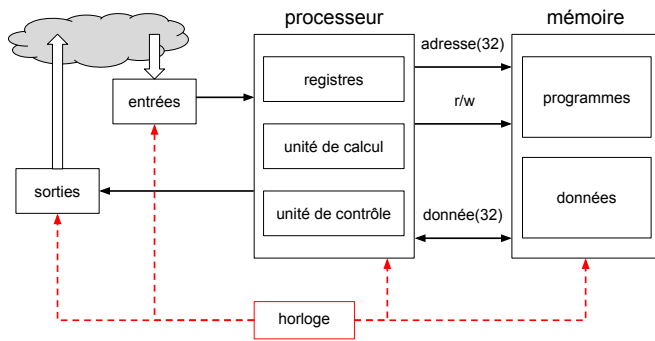
## Equipe pédagogique

Cours			
lundi	8h45-10h30	24-25-201 (provisoire)	Franck Wajsbürt
TD - TP			
<b>G1</b>	lundi	10h45-15h45	TD 24-25 — 104 TP 24-25 — 202 Franck Wajsbürt
<b>G2</b>	lundi	16h00-19h45	TD 24-25 — 108 TP 14-15 — 308 Alain Greiner
<b>G3</b>	mercredi	14h00-17h45	TD 24-25 — 108 TP 24-25 — 302 Quentin Meunier
<b>G4</b>	vendredi	14h00-17h45	TD 14-24 — 108 TP 24-25 — 302 Emmanuelle Encrenaz et Pirouz Bazargan

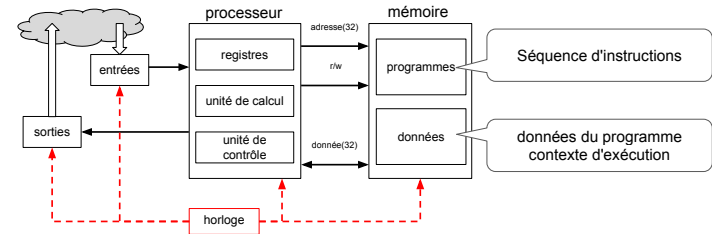
## Calendrier

sem	Cours	TD/TP	Interro
37	<b>C0</b> Vue d'ensemble du module		
38	<b>C1</b> Assembleur MIPS	<b>TD1</b> Programmation Assembleur <b>TP1</b> Présentation XSPIM	
39	<b>C2</b> Utilisation de la pile & appels de fonctions	<b>TD2</b> Appels de fonctions <b>TP2</b> Programmation fonctions / XSP:IM	
40	<b>C3</b> Interruptions / Exceptions /Trappes + GCC	<b>TD3</b> Programme complet <b>TP3</b> Génération de code GCC / XSPIM	
41	<b>C4</b> Bus système et périphériques / GIET	<b>TD4</b> Architecture mono-processeur <b>TP4</b> Génération de code pour SoC Lib	Assemb
42	<b>C5</b> Hiérarchie mémoire et techniques de cache	<b>TD5</b> Principe des mémoires cache <b>TP5</b> Effets de cache	
43	<b>C6</b> Caches (suite) Interrupts, exceptions, trappes	<b>TD6</b> Performances caches <b>TP6</b> Perfs caches	
44	Vacances		
45	Partiel corrigé		
46	<b>C7</b> Accès aux périphériques : ICU, TTY, TIMER	<b>TD7</b> Analyse GIET <b>TP7</b> Communications par interruptions	Cache
47	<b>C8</b> Périphériques DMA : contrôleurs disque et réseau	<b>TD8</b> Périphériques DMA <b>TP8</b> contrôleurs IOC, DMA et FB	
48	<b>C9</b> Multiplexage temporel & Commutation de tâches	<b>TD9</b> Commutation de tâches <b>TP9</b> Commutation de tâches	Giet
49	<b>C10</b> Principe Mémoire Virtuelle et MMU/TLB	<b>TD10</b> Partage du bus système <b>TP10</b> Architecture multi-processeur	
50	<b>C11</b> Programmation parallèle multi-tâches et synchro	<b>TD11</b> mémoire virtuelle <b>TP11</b> Rattrapage	Switch
51	Révisions guidées		
52	Vacances		
1	Vacances		
2	1ère Session		

## Architecture (ordinateur) de Von Neumann

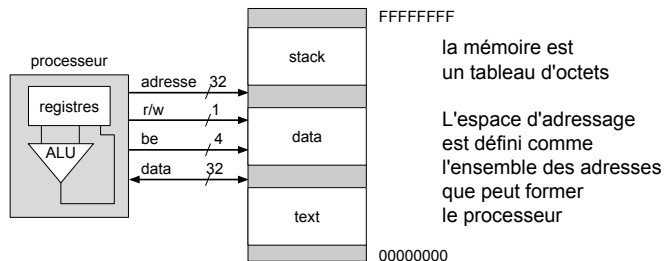


## Architecture de Von Neumann



- La mémoire stocke le programme, ses données et le contexte d'exécution.
- Des contrôleurs d'entrée-sorties permettent d'accéder à l'extérieur.
- L'unité de contrôle séquence l'exécution des instructions du programme.
- L'unité de calcul exécute des opérations arithmétiques et logiques.
- Les registres sont de la mémoire utilisés par l'unité de calcul.
- Une horloge cadence le fonctionnement, p. ex. une instruction par cycle.

## Espace d'adressage du processeur



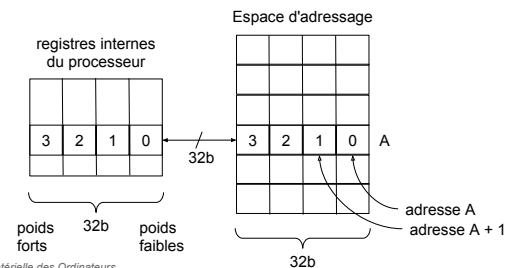
la mémoire est un tableau d'octets

L'espace d'adressage est défini comme l'ensemble des adresses que peut former le processeur

- Les adresses font 32 bits ce qui signifie 4 GiB
  - $1\text{kiB} = 2^{10}$   $1\text{MiB} = 2^{20}$   $1\text{GiB} = 2^{30}$
  - $4\text{GiB} = 2^{32} = 2^{(2+10+10+10)} = 4 * 2^{10} * 2^{10} * 2^{10} = 4\text{Gibi Bytes}$
- Chaque octet a une adresse mais les échanges sont par mots de 4 octets.
- L'ALU (boite à opérations) travaillent sur des nombres de 32 bits  
⇒ le MIPS est une architecture 32 bits
- Les registres sont donc aussi sur 32 bits.

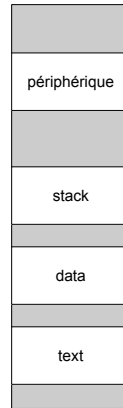
## Endianness

- Les registres font 32 bits donc un mot de 4 octets ( 3 [fort] - 2 - 1 - 0 [faible] )
- Quand on veut désigner un mot en mémoire, on donne toujours l'adresse du premier octet (celui dont l'adresse est la plus petite).
- Il existe deux conventions : **little endian** et **big endian**
- Convention **little endian**
  - L'octet de poids faible d'un mot est rangé à l'adresse la plus petite en mémoire
- Convention big endian : l'octet de poids fort est à l'adresse la plus petite



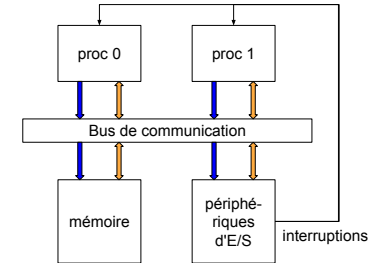
## La mémoire n'est pas l'espace d'adressage

- L'espace d'adressage, c'est l'ensemble des adresses que peut former le processeur;
- La mémoire, c'est le composant dans lequel sont stockés :
  - Le code du programme (les instructions)
  - Les données du programme
  - La pile d'exécution du programme
- Certaines adresses de l'espace d'adressage permettent d'adresser les registres de commandes des périphériques.
- Ces registres sont des cases de mémoires ou des "commandes" (lire ou écrire à ces adresses permet de commander le périphérique).




## Schéma de principe d'une machine

- Le but de l'UE est de comprendre le fonctionnement d'un ordinateur.
- Un ou plusieurs processeurs se partagent le même espace d'adressage et les mêmes périphériques d'entrées-sortie.
- Nous aborderons la programmation du processeur, l'accès à la mémoire, l'accès au périphérique sous l'angle matériel et logiciel.



## MIPS

- Le MIPS est de type RISC
    - RISC : Reduced Instruction Set Computer : 57 instructions.
    - Toutes les instructions sont codées sur 1 mot (4 octets).
    - Le MIPS démarre une instruction par cycle d'horloge.
- 
- Le MIPS dispose de 32 registres de travail (GPR : General Purpose Register)
  - L'accès à l'espace d'adressage se fait uniquement par des lectures et des écritures de registres GPR (pas de calcul)
  - Le MIPS contient une unité de calcul fonctionnant avec les GPR et plusieurs coprocesseurs spécialisés
    - c0 : coprocesseur du système d'exploitation
    - c1 : coprocesseur de calcul flottant
    - c2 : coprocesseur de gestion de la mémoire
  - Le processeur dispose de deux modes de fonctionnement
    - mode utilisateur avec des droits limités (instructions, registres, espace)
    - mode système (ou kernel) avec tous les droits

## Prochain cours

### Architecture externe et programmation assembleur du MIPS

- Types d'instruction
- Principe d'exécution d'une instruction
- Programmation assembleur
  - Concepts : symbole, directives, macro-instructions, ...
  - Structure d'un programme
  - Allocation des données globales
  - Programmation des boucles
- Présentation de l'application simulateur MARS